

QUESTION ONE:

{11*2=22 pts}

- Give 1 assembly statement to define 1 signed word, called **my_var**, having an initial value -55.

`my_var dw -55`

- Give 1 assembly statement to define a constant, called **daysec**, having a value $24*60*60$.

`daysec equ 24*60*60`

- Give 1 assembly statement to define an array **R5** of 1024 signed words containing no initial values.

`R5 dw 0, 0, ..., 0`

- Give 1 assembly statement to define a buffer **st_buf** that will be used to store a data read from the keyboard using function A of int 21h: a name (20 characters) and a CPR number (9 digits).

`st_buf db 29 dup(?)`

- The type of the operand used in the instruction `MOV EBX, OFFSET BUF` is:
 - a) Direct
 - b) Immediate
 - ☒ c) Indexed
 - d) Indirect
 - e) None
- The debug command used to execute at once all instructions in the loop body is:
 - a) d Loop
 - ☒ b) p
 - c) r Loop
 - d) p Loop
 - e) None
- The logical address `5F96:5CA4` is converted to physical address:
 - ☒ a) 63604
 - b) BC3A
 - c) 5F96+5CA4
 - d) 5CA40+5F96
 - e) None
- The directive `T5 word 2F, 4 dup(3 dup(6CH, 44), 'A9')` occupies _____ bytes.
 - a) 29
 - ☒ b) 58
 - c) 37
 - d) 74
 - ☒ e) None
- The _____ accepts input files with extension `.ASM` and produces output files with extension `.Lst`
 - ☒ a) ASSEMBLER
 - b) DEBUG
 - c) LINKER
 - d) COMPILER
 - e) None
- If register `ax` contains `A4CE`, then after executing `"ROR ax, 4"` register `ax` will contain:
 - a) 0A4C
 - b) 4CEA
 - ☒ c) EA4C
 - d) CEA4
 - e) None
- If register `ax` contains `A4CE`, then after executing `"SAR ax, 4"` register `ax` will contain:
 - a) 0A4C
 - b) 4CEA
 - c) CEA4
 - ☒ d) FA4C
 - e) None

QUESTION TWO:

{5+5=10 pts}

- Give a sequence of instructions to calculate: $A = (B \times C) \times D$, where B, C, and D are all memory words already initialized with proper signed values, and A is defined as a doubleword.

<pre> mov ax, B mul ax, C mov ax, D mul ax mov eax, ax </pre>	
---	--

- Give a sequence of instructions that stores in BL register the count of "ZERO" bits in EAX register and stores in BH register the count of "ONE" bits in EAX register. (Use single loop).

<pre> mov bl, 0 mov bh, 0 mov ecx, eax L1: shr ecx, 1 jc L1 inc bl jnz L1 </pre>	<pre> L2: shr ecx, 1 jnc L2 inc bh jnz L2 </pre>
--	--

QUESTION THREE: Convert the following C++ code into equivalent assembly code {12 pts}

```
int num, sum = 0;
cout << "Enter a number please: ";
cin << num;
if (num < 0)
    num = -num;
while (num != 0)
{
    sum += num % 10;
    num /= 10;
}
cout << "Sum = " << sum << endl;
```

Include <iostream>

using namespace std;

int main()

{

int num, sum = 0;

cout << "Enter a number please: ";

cin >> num;

if (num < 0)

{

num = -num;

while (num != 0)

{

sum += num % 10;

num /= 10;

}

cout << "Sum = " << sum << endl;

}

return 0;

}

WWW.UOBVIEW.COM

Uploaded By Smart Error

QUESTION FOUR:

{4+4+4=12 pts}

- a) Give **NO MORE THAN 4** instructions to subtract $M3 = M3 - M4$, where $M3$ and $M4$ are quadword memory locations (8 bytes each) defined as shown below:

M3 Qword 70F05060D0801020H
M4 Qword 6020E03090C010F0H

- b) Give **NO MORE THAN 4** instructions to swap the left half of EAX register with the right half of EBX register and swap the right half of EAX register with the left half of EBX register.

- c) Carefully study the following code and give **ONE** phrase strictly describing what does it do?

```
mov ecx, eax
Ror ebx, 16
shld eax, ebx
```

```
.data
T1 BYTE "ITCS241, Computer, Science, College, of, IT, University, of, Bahrain", 0
.code
MOV     EDX, offset T1
XOR     EAX, EAX
MOV     ECX, lengthof T1
L2: cmp     byte ptr [EDX], ','
JNE     L3
INC     EAX
MOV     byte ptr [EDX], ';'
L3: INC     EDX
LOOP    L2
CALL    WRITEDEC
```

QUESTION FIVE:

{2+2+5+3=12 pts}

- a) Give 1 instruction to store 0 in CX register without using MOV instruction.

`xor cx, cx`

- b) Give 1 instruction to add the value 2AC9H to the contents of memory word pointed by BX register.

`add word ptr [bx], 2AC9`

- c) Give NO more than 5 instructions to compute: $EAX = AL * 1024 - BL * 128$. Using MULTIPLY instructions is NOT allowed.

```
movzx eax, al
shl eax, 10
movzx ecx, bl
shl ecx, 7
sub ecx, eax
```

- d) Give no more than 3 instructions to divide the predefined 2 signed byte values: BT1 / BT2.

```
movzx eax, bt1
movzx ecx, bt2
div ecx
```

QUESTION SIX: USE INTERRUPTS FOR ALL INPUT / OUTPUT OPERATIONS. {12 pts}

Write a procedure **macP8** that accepts a letter passed in **dI** register and displays on the screen the letter itself and its left and right neighboring letters. For example, if you pass 'f', the procedure will display 'efg'.

Write a complete program that: prompts the user to enter from the keyboard a lower-case letter (b...y), validates the input, loops until a valid letter is entered, and calls the developed above procedure **macP8**.

```

- model: 386
- stack: 100h
- data:
    prompt db "Enter lower case letter: ", 13, 10, '$'
- code:
    main proc
V: mov     AH, 09h
    mov     DX, offset prompt
    int     21h

    mov     AH, 0Ah
    int     21h
    mov     AL, 0Ah
    mov     CX, 0Fh
L:     cmp     AL, 0Fh
    jz     done
    inc     AL
    loop    L
done:
    call    macP8
    mov     AH, 09h
    int     21h
    mov     endp

macP8 proc
    push    DX
    mov     CX, 0Fh
    mov     EB, 02h
    mov     EB, (AL-1)
    int     21h

```


QUESTION SEVEN:

{2x4=08 pts}

Study carefully the following program and answer all questions below

```

1) INCLUDE I RVINE32.INC
2) SHIFT MACRO VAL, AMT, DIR
3) SH&DIR VAL, AMT
4) ENDM
5) .CODE
6) MAIN PROC
7) XOR EAX, EAX
8) MOV BL, 20H
9) MOV BH, 0
10) MOV AX, BX
11) SHIFT AX, 6, L
12) SHIFT BX, 2, L
13) SUB AX, BX
14) MOVZX EAX, AX
15) CALL WRITEHEX
16) MOV DX, 3F7CH
17) SHIFT DX, 4, R
18) EXIT
19) MAIN ENDP
20) END MAIN
    
```

- The expansion of the macrocall in line 11 is:

SHL AX, 6

- After executing the above program, the value displayed is 00000080 H

- The expansion of the macrocall in line 17 is:

SHR DX, 4

- After executing the above program, the register DX will contain 03F7 H.